

Prepared by Mohammad Umer

Python Tutorial

2.1 - Operations on Variables

Following are some common operators in python:

1. Arithmetic operators: +, -, *, /, //, %
2. Assignment operators: =, +=, -=, *=
3. Comparison operators: ==, >, >=, <, !=
4. Bitwise operator: &, |, ~, ^, <<, >>
5. Logical operators: and, or, not.
6. Membership operator: in, not in
7. Identity Operator: is, is not

1. Arithmetic Operators

```
In [8]: a=10
b = 5

print("Addition : ",a+b)
print("Subtraction : ",a-b)
print("Multiplication : ",a*b)
print("Division : ",a/b)           # It gives the 'Coefficient'
print("Floor Division : ",a//b)
print("Modulus : ",a%b)           # It gives the 'Remainder'

print("Exponentiation : ",a**b)    # a raised to the power of b
```

```
Addition : 15
Subtraction : 5
Multiplication : 50
Division : 2.0
Floor Division : 2
Modulus : 0
Exponentiation : 100000
```

Q2. Addition of Two numbers (initialized variables)

```
In [4]: a = 2
        b = 4

        c = a + b
        print(f"Addition: {c}")
```

Addition: 6

Different ways of writing the code

```
In [9]: a,b = 2,4
        print(f"Addition : {a} + {b} = {a+b}")
```

Addition : 2 + 4 = 6

OR using

format() method: It allows you to embed variables value into a string using placeholders defined by curly braces {}.

```
In [4]: a,b = 2,4

        c = a + b
        print("Addition : {} + {} = {}".format(a,b,c))
```

Addition : 2 + 4 = 6

Q3. Addition of Two numbers (Taking input from the User)

```
In [14]: # WRONG

        a = input("Enter value of a : ")          # character
        b = input("Enter value of b : ")

        c = a + b
        print(f"Addition : {a} + {b} = {c}")
```

Addition : 2 + 4 = 24

```
In [12]: # PROOF
        print(type(a))
```

<class 'str'>

Correct code for our Question

```
In [13]: a = int(input("Enter value of a : "))          # TYPE Conversion
        b = int(input("Enter value of b : "))

        c = a + b
        print(f"Addition : {a} + {b} = {c}")
```

Addition : 2 + 4 = 6

Q4. Write a Python program to convert Celsius to Fahrenheit.

```
In [14]: celsius = float(input("Enter temperature in Celsius: "))
        fahrenheit = (celsius * 9/5) + 32

        print(f"{celsius}°C is equal to {fahrenheit}°F")
```

25.0°C is equal to 77.0°F

Q5. Write a Python program to swap the values of two variables.

```
In [18]: a = 29
        b = 4
        print(f"a = {a} and b = {b}")

        ## Display Swapping
        print(f"a = {b} and b = {a}")
```

a = 29 and b = 4
a = 4 and b = 29
29

Actual Swapping is below

```
In [20]: a = 29
        b = 4
        print(f"a = {a} and b = {b}")

        ## Actual Swapping
        temp = a
        a = b
        b = temp
        print(f"a = {a} and b = {b}")

        ## Shorthand ---->    a, b = b, a
```

a = 29 and b = 4
a = 4 and b = 29
4

Q6. Write a Python program to solve a **Marketing Sheet Problem** where:

- The user inputs the product name, rate (price per unit), and quantity sold.
- The program calculates the total amount using the formula: $\text{Total Amount} = \text{Rate} \times \text{Quantity}$
- Finally, it displays the Bill with the total amount.

```
In [5]: # Input product details
product_name = input("Enter product name: ")
rate = int(input("Enter rate: "))
quantity = int(input("Enter quantity: "))

# Calculate total amount
total_amount = rate * quantity

# Display the bill
print("\n---- Bill ----")
print(f"Product Name: {product_name}")
print(f"Rate: {rate}")
print(f"Quantity: {quantity}")
print(f"Total Amount: {total_amount}")
```

```
---- Bill ----
Product Name: Coca cola
Rate: 25
Quantity: 2
Total Amount: 50
```

Q7. Write a Python program to solve a **Marks Card Problem** where:

- The user inputs the student name, marks in subjects (English, Math, Science, Social Science, Urdu).
- The program calculates the total marks and the percentage.
- Finally, it displays the Marks card of the student.

```
In [ ]: # Input student details
student_name = input("Enter student name: ")
print("Enter marks for the following subjects:")

# Input marks for each subject
english = int(input("English: "))
math = int(input("Math: "))
science = int(input("Science: "))
social_science = int(input("Social Science: "))
urdu = int(input("Urdu: "))

# Calculate total marks and percentage
total_marks = english + math + science + social_science + urdu
percentage = (total_marks / 500) * 100

# Display the marks card
print("\n---- Marks Card ----")
print(f"Student Name: {student_name}")
print(f"English: {english}")
print(f"Math: {math}")
print(f"Science: {science}")
print(f"Social Science: {social_science}")
print(f"Urdu: {urdu}")
print(f"Total Marks: {total_marks}/500")
print(f"Percentage: {percentage:0.2f}%")           # 91.6% ----> 91.60
```

Enter marks for the following subjects:

```
---- Marks Card ----
Student Name: Mohammad Umer Jan
English: 99
Math: 97
Science: 96
Social Science: 86
Urdu: 85
Total Marks: 463/500
Percentage: 92.60%
```

format specifier : It is a set of instructions used to define how a value (like a number, string, or date) should be formatted and inserted into and shown as part of a text that is printed or outputted.

Syntax: `{[field_name]:[format_spec]}`

Specifier	Description	Example	Output
<code>d</code>	Decimal (integer).	<code>f"{42:d}"</code>	42
<code>.nf</code>	Fixed-point with <code>n</code> decimal places.	<code>f"{3.14159:.2f}"</code>	3.14
<code>e</code>	Scientific notation with lowercase <code>e</code> .	<code>f"{12345:e}"</code>	1.234500e+04
<code>%</code>	Percentage (multiplies by 100 and adds <code>%</code>).	<code>f"{0.75:%}"</code>	75.000000%
<code>,</code>	Adds thousands separator.	<code>f"{1234567:,}"</code>	1,234,567
<code>x</code> or <code>X</code>	Hexadecimal (lowercase or uppercase).	<code>f"{255:x}"</code>	ff

2. Assignment Operators

```
In [1]: x = 10
print("Initial value of x:", x)

x += 5
print("After x += 5:", x)

x -= 3
print("After x -= 3:", x)

x *= 2
print("After x *= 2:", x)

x /= 4
print("After x /= 4:", x)
```

```
Initial value of x: 10
After x += 5: 15
After x -= 3: 12
After x *= 2: 24
After x /= 4: 6.0
```

3. Comparison Operators

```
In [2]: a = 7
        b = 5

        print("a == b:", a == b)
        print("a != b:", a != b)
        print("a > b:", a > b)
        print("a >= b:", a >= b)
        print("a < b:", a < b)
        print("a <= b:", a <= b)
```

```
a == b: False
a != b: True
a > b: True
a >= b: True
a < b: False
a <= b: False
```

4. Bitwise Operators

```
In [7]: a = 5          # Binary: 0101
        b = 3          # Binary: 0011

        print("Bitwise AND (a & b):", a & b)
        # Only bits that are 1 in both numbers stay 1
        # 0101 & 0011 = 0001 → Result: 1

        print("Bitwise OR (a | b):", a | b)
        # Any bit that is 1 in either number becomes 1
        # 0101 | 0011 = 0111 → Result: 7

        print("Bitwise NOT (~a):", ~a)
        # Flips all bits of a and gives the two's complement
        # ~a = -a - 1 = -5 - 1 = -6

        print("Bitwise XOR (a ^ b):", a ^ b)
        # Bits are 1 only if they are different in a and b
        # 0101 ^ 0011 = 0110 → Result: 6

        print("Left Shift (a << 1):", a << 1)
        # Shifts bits of a left by 1 place → 0101 becomes 1010
        # Result: 10 (Equivalent to 5 * 2^1)

        print("Right Shift (a >> 1):", a >> 1)
        # Shifts bits of a right by 1 place → 0101 becomes 0010
        # Result: 2 (Equivalent to 5 // 2)
```

```
Bitwise AND (a & b): 1
Bitwise OR (a | b): 7
Bitwise NOT (~a): -6
Bitwise XOR (a ^ b): 6
Left Shift (a << 1): 10
Right Shift (a >> 1): 2
```

5. Logical Operators

```
In [4]: x = True
        y = False

        print("x and y:", x and y)
        print("x or y:", x or y)
        print("not x:", not x)
```

```
x and y: False
x or y: True
not x: False
```

6. Membership Operators

```
In [ ]: list1 = [1, 2, 3, 4, 5]

        print("3 in list1:", 3 in list1)
        print("6 not in list1:", 6 not in list1)
```

```
3 in list1: True
6 not in list1: True
```

7. Identity Operators

```
In [ ]: a = [1, 2]
        b = a
        c = [1, 2]

        print("a is b:", a is b)           # True, same object
        print("a is c:", a is c)           # False, different objects
        print("a is not c:", a is not c)   # True
```

```
a is b: True
a is c: False
a is not c: True
```


2.2 - Control Flow Statements

These statements control the flow of execution in a program, allowing for alterations in the normal sequence of execution.

Types of Control Flow Statements:

1. **Conditional Statements:** `if`, `elif`, `else`

- Used to execute code based on a condition.

2. **Looping Statements:** `for`, `while`

- Used to repeatedly execute a block of code.

3. **Jump Statements:** `break`, `continue`, `return`

- Used to alter the flow of control within loops or functions.